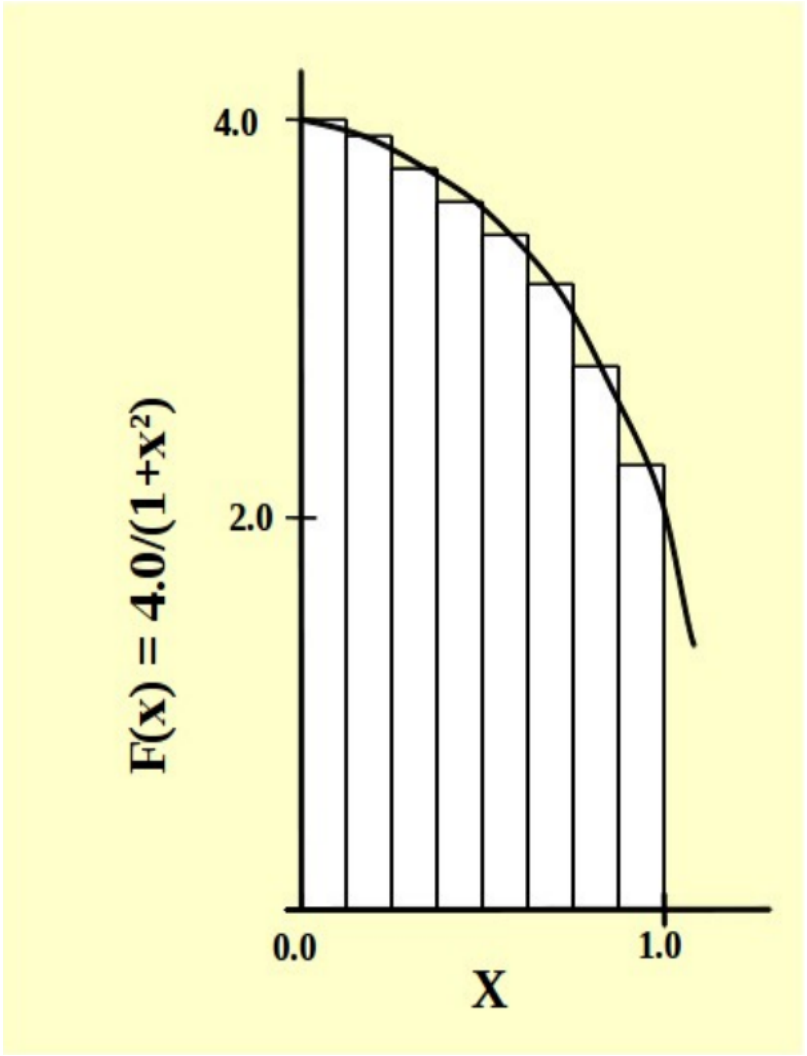
UNIL | Université de Lausanne

Exercise sheet

Simon Scheidegger
simon.scheidegger@unil.ch
September 27th, 2018
Cowles Foundation – Yale

Supplementary material for the exercises is provided in YaleParallel2018/day4/code/supplementary_material

Recall: compute Pi



Mathematically, we know that:

$$\int_0^1 \frac{4.0}{(1+x^2)}\,dx = \pi$$

We can approximate the integral as a sum of rectangles:

$$\sum_{i=0}^{N} F(x_i)\Delta x \approx \pi$$

Where each rectangle has width Δx and height $F(x_i)$ at the middle of interval i.

1. Hybrid computaton of Pi

1. go to YaleParallel2018/day4/code/supplementary_material

> cd YaleParallel2018/day4/code/supplementary_material

2. Have a look at the source code in Fortran/CPP that compute Pi OpenMP parallel.

3. Make the code hybrid parallel by adding MPI.

4. write a makefile that compiles the code in hybrid.

5. run the code with a slurm file in hybrid mode. Experiment with the settings/distribution of MPI processes and OpenMP threads.

   → **2 MPI processes & 10 OMP threads, 4 MPI & 5 Openmp, ...**

2. Scaling test – Discrete State DP

Go to the massively parallelized DSDP code (set nk = 36,000 – a big number)

YaleParallel2018/day4/code/DP_MultComms

→ perform a fully-fledged hybrid scaling test.
→ 1 thread, 1 MPI process.
→ 1 node, 1 MPI process – max. threads/node (hybrid – comm not split).
→ 5 nodes (comm split).
→ 10 nodes (comm split).
→ 20 nodes (comm split).
(see how much you can request – be careful with the wall time requested)

→ **all on GRACE – please use the slurm file that allows for hybrid jobs.**

→ **Generate scaling plots, normalized to**
a) one CPU.
b) one Node.